

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
29 November 2001 (29.11.2001)

PCT

(10) International Publication Number
WO 01/91417 A2

(51) International Patent Classification⁷: **H04L 29/06**

[US/US]; 1224 Mellow Lane, Simi Valley, CA 93065 (US).

(21) International Application Number: PCT/US01/16329

BOUDREAU, Paul [US/US]; 110 Harvard Street, No 1, Cambridge, MA 02138 (US).

(22) International Filing Date: 21 May 2001 (21.05.2001)

(74) Agents: **ECCLESTON, Lynn, E.** et al.; Pillsbury Winthrop LLP, 1100 New York Avenue, N.W., Washington, DC 20005 (US).

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/205,987 19 May 2000 (19.05.2000) US
09/664,724 19 September 2000 (19.09.2000) US

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier applications:

US 60/205,987 (CIP)
Filed on 19 May 2000 (19.05.2000)
US 09/664,724 (CIP)
Filed on 19 September 2000 (19.09.2000)

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

(71) Applicant (*for all designated States except US*): **BROAD-STREAM.COM, INC.** [US/US]; 21300 Victory Boulevard, Suite 1100, Woodland Hills, CA 91367 (US).

Published:

— *without international search report and to be republished upon receipt of that report*

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **KAUFMAN, John** [US/US]; 7810 Topanga Cyn Blvd. #102, Canoga Park, CA 91304 (US). **KWON, Young, D.** [KR/US]; 22015 Lemarsh Street, Chatsworth, CA 91311 (US). **ROGERS, Tony, J.**

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: MANAGEMENT AND DELIVERY OF ONLINE WEBCASTS

(57) Abstract: A streaming media delivery system employs multiple client data networks storing copies of the streaming media. When a viewer wishes to see content, the system chooses a client data network which is best for providing the media content and directs a client wrapper object on the viewer's system to that network. When the quality of delivery from that network becomes too low, a client wrapper object on the viewer's system can request a switchover to a new client data network. The system may also redirect the wrapper object to receive content from a different network for maintenance purposes and the like. The viewer wrapper object also provides monitoring information on communication line quality and the like to the system for feedback and logging purposes.



WO 01/91417 A2

MANAGEMENT AND DELIVERY OF ONLINE WEBCASTS

RELATED APPLICATIONS

This application is based on and claims priority under 35 U.S.C. § 120 from U.S. Provisional Patent Application No. 60/205,987 filed May 19, 2000 and
5 U.S. Application Serial No. 09/664,724 filed September 19, 2000 entitled
“Management and Delivery of Online Webcasts”, the contents of which are fully incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention is directed to systems for performing online delivery of content over distributed communications networks. More specifically, the invention is directed to systems for delivering streaming media content over the Internet.

2. Background of the Related Art

The Internet and World Wide Web (WWW) have proven to be valuable tools for delivering multimedia to users. In particular, the Internet is becoming effective at delivering streaming multimedia content to a large number of users. When the number of users receiving the streaming media is small, prior art
20 systems are sometimes capable of providing a smooth, jerk-free presentation. However, prior art systems have had difficulty accommodating very large numbers of users. When the number of users is particularly large, the system

becomes slow, provides error-filled, jerky streams and generally exhibits unacceptable performance characteristics.

SUMMARY OF THE INVENTION

The present invention has been made in view of the above and other
5 problems of the prior art. The present invention provides a streaming media distribution system which can accommodate a large number of clients while providing high-quality streaming media to those viewers.

The above objects are achieved according to a first aspect of the present invention by providing a streaming media delivery system which employs
10 multiple data networks storing copies of the streaming media. When a viewer wishes to see content, the system chooses a client data network which is best for providing the media content and directs a client wrapper object on the viewer's system to that network. When the quality of delivery from that network becomes low, the client wrapper object on the viewer's system can request a switchover to
15 a new *client data* <streaming network> network. The system may also redirect the wrapper object to receive content from a different network for maintenance purposes and the like. The viewer wrapper object also provides monitoring information on communication line quality and the like to the system for feedback and logging purposes.

20 BRIEF DESCRIPTION OF THE DRAWINGS

These and other objects, features and advantages of the present invention are better understood by reading the following detailed description of the preferred embodiment, taken in conjunction with the accompanying drawings, in which:

FIGURE 1(a) shows an example of content streaming architecture/system and network according to a preferred embodiment of the present invention;

FIGURE 1(b) depicts aspects of the content acquisition, publishing, distribution and delivery system according to the present invention.

5 **FIGURES 2(a)** and **2(b)** are block diagrams of a client computer and a network server, respectively, in accordance with embodiments of the present invention;

FIGURE 3 is a flowchart summarizing the process of a client obtaining a reference file according to embodiments of the present invention;

10 **FIGURE 4** is a state diagram of a client according to embodiments of the present invention;

FIGURE 5 is a flowchart depicting operation of the redirection and bandwidth managers according to embodiments of the present invention;

15 **FIGURE 6** is a state diagram of a monitoring manager according to embodiments of the present invention;

FIGURE 7 is depicts operations of the monitoring manager according to embodiments of the present invention;

20 **FIGURES 8(a)** and **8(b)** show the inclusion of advertisement in different types of streaming content delivery according to embodiments of the present invention; and

FIGURE 9 is a flowchart of typical operations of systems according to embodiments of this invention.

DETAILED DESCRIPTION OF PRESENTLY PREFERRED EXEMPLARY EMBODIMENTS

25 **FIGURE 1(a)** shows a content streaming architecture/system and network, generally designated by the reference numeral **100**, according to a preferred

embodiment of the present invention. A number of network clients **102-1**, **102-2**, ..., **102-n** (collectively, network clients **102**) are connected to the public distributed computer network known as the Internet **104** via their respective Internet Service Providers (ISPs) **106** (only one ISP is shown in the drawing) in a manner known in the art. In the embodiments shown, the system **100** is implemented over the public Internet **104**, using the World Wide Web (WWW or Web) and its hyperlinking capabilities. The invention is applicable to other networks as well, although it is particularly suited to networks that are configured to allow hyperlinks. In addition, the invention is also applicable to downloading, wireless streaming and so on.

FIGURE 2(a) shows an example of a client computer **102**. Various types of network clients **102** can be utilized, such as palmtop computers, notebook computers, personal organizers, and the like. Client computer **102** includes conventional components, including a data processor, central processing unit (CPU) **108**; volatile and non-volatile primary electronic memory **110**; secondary memory **112** such as hard disks, floppy disks and/or other removable media; network interface components **114**; display devices interfaces and drivers **146**; audio recording and rendering components **118**; and other components as are common in personal computers. Generally, the invention operates with / on any portable devices, including any device that can download and play content.

Network clients **102** are preferably configured with a consumer-oriented operating system such as one of Microsoft Corporation's Windows operating systems, referenced in **FIGURE 2(a)** by numeral **120**. In addition, network client **102** runs an Internet browser **122** such as Netscape™ Communicator or Microsoft's Internet Explorer.

Network client **102** also includes one or more streaming multimedia data players or rendering components **124**. This software component(s) is capable of establishing streaming data connections with Internet servers or other servers, and of rendering the streaming data as audio or video. Specifically, player **124** is
5 configured to open reference files (described below) and of establishing streaming data connections with the network resources specified in the reference files, using the network transport protocols specified in the reference files, or in conjunction with the reference file. The reference files are stored on computer-readable storage media of servers or other network sources or are generated, as described
10 below. In preferred embodiments of this invention, the player **124** is preferably configured to open reference files such as ASX files (files having a filename extension of "asx"), or ASF files (files having a filename extension of "asf"),

Player(s) **124** can be implemented as a standalone component, program or as a component control such as OCX control (OCX controls are standard features
15 of programs designed for Windows operating systems). In either case, it is registered with the operating system so that it is invoked to open the appropriate reference files (ASX files and the like) in response to user requests. In the Microsoft Windows operating system, such a user request is made by double clicking on an icon representing a reference file. From within an Internet browser,
20 the request for a reference file is made by selecting (e.g., single-clicking on) a hyperlink contained in a hyperlink document that is being displayed. When this happens, the player is loaded and executed, and the subject reference file is provided to the player as a run-time argument.

Each client **102** is or includes some form of computer or similar device
25 capable of accessing the distributed computer network known as the Internet **104**

via the client's respective ISP **106** using a typical browser **122**. A client **102** may also be a dedicated Internet terminal, a device accessing the Internet using a television set and the like. Clients **102** may have any type of access to their respective ISPs **106** using network interface **114**. E.g., clients may connect to their ISPs via modems, other networks, cable television systems and the like. Each ISP **106** is connected in some manner to the Internet **104**, as is known in the art.

In some of the following description, various entities such as the ISPs **106** and the Internet **104** are omitted and all data flow is described directly between the various entities. However, the omission of entities such as the ISPs and the Internet in the description are merely to aid and simplify explanation of the system **100**. Actual data flow between entities, e.g., a client **102-1** and the redirection manager **112** would take place via the client's ISP **106** and the Internet **104**.

Also connected to the Internet **104** is a web host **126** serving a web site **128** to users over the Internet **104**. Clients **102** may view the contents of the web site **128** using an Internet browser **122** in a manner known in the art. For example, a client **102** may view the web site **128** using a Netscape™ browser running on the client. The web site **128** may offer, for example, streaming media content which the client may access using the preferred embodiment.

The content streaming architecture/system **100** also includes various server computers. An example of a server computer is illustrated in block form in **FIGURE 2(b)**. Generally, a server computer includes conventional components similar to those of network client **102** such as a data processor (CPU) **130**; volatile and non-volatile primary electronic memory **132**; secondary memory **134** such as

hard disks and floppy disks or other removable media; network interface components 136; display devices interfaces and drivers 138; and other components that are well known. The server computer runs an operating system 140 such as the Windows NT operating system from Microsoft Corporation.

5 Network servers and their operating systems are configured in accordance with known technology so that they are capable of streaming data connections with clients. The network servers generally support various network transport protocols, including multicast UDP/IP and unicast protocols such as UDP/IP, TCP/IP, and HTTP. The servers include storage components (such as secondary
10 memory 134), on which various data files are stored, formatted appropriately for efficient transmission using the noted protocols. Compression techniques are preferably used to make the most efficient use of limited Internet bandwidth.

 In the case of both network servers and client computer, the data processors are programmed by means of instructions stored at different times in
15 the various computer-readable storage media of the computers. Programs are typically distributed, for example, on floppy disks or CD-ROMs. The programs may be compiled or interpreted or, if appropriate, distributed and/or executed in some other manner. From the distribution media, the programs are installed or loaded into the secondary memory of a computer. At execution, they are loaded
20 at least partially into the computer's primary electronic memory. The invention described herein includes these various types of computer-readable storage media when such media contain instructions or programs for implementing the described steps in conjunction with a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and
25 techniques described below.

For purposes of illustration only, programs, program components and other mechanisms are shown in **FIGURES 2(a)** and **2(b)** as discrete blocks within a computer, although it is recognized that such programs, components and mechanisms reside at various times in different storage components of the computer.

The content streaming architecture/system **100** according to embodiments of the present invention also includes at least one redirection server **142** having a redirection manager **143** and at least one monitoring server **144** having a monitoring manager **145**. The redirection server **142** (also referred to herein as the redirector) and the monitoring server **144** are both connected to the Internet **104**. In some embodiments, the redirection manager **143** and monitoring manager **145** may be collocated on the same server or they may be on different servers. That is, in some cases the redirection server **142** and the monitoring server **144** may be the same physical server.

In addition, the content streaming architecture/system **100** according to preferred embodiments of the present invention also includes a number of content distribution networks (CDNs) **146-1, 146-2, ..., 146-k** (collectively referred to as CDNs **146**). In preferred embodiments, some or all of the CDNs **146** preferably hold the same streaming media content so as to provide a degree of fault tolerance and data redundancy to the system. The manner in which data (content) is acquired and distributed to CDNs **146** will be described below with reference to the content storage and distribution system **148**.

A personalization server **150**, client database **152**, real-time bandwidth distribution database **154**, CDN database **156** and an advertising database **158** are

also accessible to the redirection manager 143 and to the monitoring manager 145, either directly or via other servers. The personalization server 150 accesses the client database 152 which contains information about particular clients and is used, e.g., by the redirection manager 143 to customize a particular client's access. The advertising database 158 contains available advertising content which can be inserted into the content selected by the client. The manner in which the advertising database 158 is populated by an advertising storage and distribution system is described below. The monitoring manager 145 also has access to a real-time logging database 160.

FIGURE 1(b) depicts aspects of the content acquisition, publishing, distribution and delivery system according to the present invention. As shown in FIGURE 1(b), various content providers including Fox 168-1, CBS 168-2, NBC 168-3, ..., and PBS 168-q (collectively 168) provide streaming content in the form of scheduled broadcast events and video on demand events. The content schedule and other information is stored in a electronic program guide (EPG) database 164 in a known manner.

In addition, a content database 170 contains the actual content obtained in various ways from the content providers 168. For example, the content database 170 may be populated using a so-called crawler 172 which, along with channel definition information obtained from the EPG database 164, automatically uploads content to the content database 170. Further, content providers 168 can, if they so choose, push their content to the content database 170 using content pushing. A storage manager 174 handles requests for storage and retrieval of content data to/from the content database 170.

In some preferred embodiments of the present invention, advertising is inserted in or included with streamed content. The adverts are stored in and taken from an advertising database 158. An ad manager 174 takes ads from the advertising database 158 for inclusion in a particular stream. In some preferred
5 embodiments, advertising is included in the advertising database 158 based on a real-time advertising auction. Various advertising agencies or advertisers (e.g., DoubleClick 176-1, Engage 176-2, and other conventional/traditional broadcast advertising agencies, collectively ad. agencies 176) provide advertising to the advertising database 158, in some cases via an advertising auction, preferably a
10 real-time auction.

In the auction, various of the ad agencies 176 bid on advertising spots at a real-time ad auction site. Each advertising agent 176 maintains an account with the auction site and maintains credit (stored in the advertiser / finance database 178) The auction site obtains the available time spot information from the EPG
15 database 164.

Once bidding starts for a particular time spot, the highest bids per spot are noted. Bidding for a particular time spot is closed before air time, preferably two (2) minutes before air time. For each bid, a check is made at the advertiser / finance database 178 to determine whether or not the bidder has sufficient funds
20 or credit to pay for the bid. Ad agencies 176 can purchase additional credits online while they are bidding.

While the auction is taking place, all bids are ranked and all bidding parties are shown a count down to the end of bidding and the current bids.

When the bidding closes, the ad with the highest bid is stored in the advertising database 158 for use at the appropriate time. When the ad is included
25

in a reference file (and preferably when actually played), the cost of the ad is automatically deducted from the winning agency's available credit in the advertiser / finance database 178.

5 The operation of the system 100 is described now with reference to FIGURES 1(a)-6.

Content is sent from the content database 170 to the various CDNs 148, based on various distribution rules and procedures. Multiple CDNs 148 are used to achieve data redundancy. This means that data (content) are stored in more than one CDN distribution point, thereby providing fault tolerance.

10 In some preferred embodiments, instead of storing the same information at all CDNs 148, the information is stored at only some (e.g., 20% to 40% of the CDNs). E.g., if there are a total of 10 or 12 CDNs, data are stored at 2 to 4 CDNs. This approach provides redundancy while minimizing the cost of storage at CDNs. CDN storage cost is a factor applied in load balancing of the system.

15 In preferred embodiments, twenty percent of the content in the content database 170 is stored to serve 80% of the traffic.

In operation, when a client 102 chooses to view streaming media content from the web site 128, the client uses browser 122 to display the contents of the web site 110 on some form of display (not shown) such as a computer monitor using display components 138. Each program or event which a client might wish
20 to play (view, hear, etc) has a corresponding associated identity (*ProgramId* or *EventId*, respectively). Also, each client (or user) has a corresponding associated identity (*ClientId*). When client requests specific content from the web site 128, the client's Internet browser 122 provides both the client's identity (*ClientId*) and
25 the requested program or event identity (*ProgramId* or *EventId*).

In some aspects, the present invention incorporates an object wrapper (monitor) program (described below). In preferred embodiments of the present invention, a client **102** needs an object wrapper (monitor) program to be loaded onto the client **102**. Accordingly, the first time a particular client **102** accesses the web site **128** in order to display streaming media content, a client object wrapper (monitor) **162** is downloaded automatically to the client **102**. The client object wrapper module **162** may be downloaded using a well-known HTML technique such as the following:

```

10      <OBJECT classid="CLSID:931EFE2D-1F50-11D4-A821-
      00508B8D40C9" codeBase=prjWmpClient.CAB#version=1,0,0,0
      id=wmpClientCtl style="LEFT: 0px; TOP: 0px"
      VIEWASTEXT>
      <PARAM NAME="_ExtentX" VALUE="8625">
      <PARAM NAME="_ExtentY" VALUE="6509">
15      <PARAM NAME="FILENAME"
      VALUE="http://fusion.broadstream.com/screenblaster/MakePlaylist.as
      p?CDN=0&FILE_NAME=bats-3w-filmclip1.asf&.asx"></OBJECT>

```

Once the wrapper module **162** has been downloaded onto the client **102**, and when the web page from the web site **128** is loaded into client's system, the client's browser **122** is redirected to the redirection manager **143** on the redirection server **142**. The client's browser **122** provides the redirection manager **143** with information allowing the redirection manager to redirect the client's request appropriately, according to the present invention. In preferred embodiments of this invention, the browser **122** provides the redirection manager **143** with the both the client's identity (*ClientId*) and the requested program or event identity (*ProgramId* or *EventId*). That is, when the client **102** selects (e.g., clicks on) a link on web site **110** in order to obtain streaming content (e.g., audio, video, download) therefrom, the client's request goes to redirection manager **143**. The redirection manager **143** chooses (in a manner described below) a particular CDN **146** to serve that requested media content to the requesting client **102**. Once

the redirection manager **143** has determined which particular CDN **146** the client **102** is to use, the redirection manager **143** then returns information back to the client **102** so that the client can immediately view the requested content. The returned information from the redirection manager **143** is preferably a reference file, e.g., an ASF (Advanced / Active Streaming Format), ASX, RM, SMIL (Synchronized Multimedia Integration Language) or other type of content data. In a preferred embodiment, the reference file sent from the redirection manager **143** to the client **102** is an ASX file. For the sake of explanation only, and without loss of generality, the file returned by the redirection manager **143** to the client **102** will be referred to herein interchangeably as the reference file and as the ASX file.

A reference file is a text file that links Web pages to streaming media-based content on a media server. The reference file generally redirects streaming media content away from the client's browser **103** to the appropriate media player mechanism **124** running on the client. In addition, in the present invention, the reference file also redirects the client's media player mechanism **105** to the appropriate selected CDN **146**. When the client's browser **103** downloads a reference file, the type of file (determined, e.g., from the file's name extension) causes the browser **103** to invoke the appropriate media player mechanism **105** which then locates and plays the content specified in the file.

Reference files are generally integrated into the WWW. Hyperlinks to the reference files are placed in Web documents, and a user retrieves a particular reference file by clicking on its hyperlink. In response, the user's Internet browser **122** retrieves the reference file from the network source and opens it with the appropriate player **124**, depending on the type of reference file. Player **124**, in

turn, uses the reference file to establish a streaming data connection which the player then renders.

A typical reference file generally includes a plurality of lines, each containing a different resource specifier in standard network Uniform Resource
 5 Locator (URL) format. The order of the resource specifiers establishes a preferred order for attempting communications with the resources specified by the resource specifiers. Each resource specifier is preceded, in some protocols, by an identifier of the form "Ref#=URL", where the # portion of the identifier is a number which indicates the preferred order for attempting communications. For
 10 example, *Ref1* is before *Ref2*. Alternatively, the reference file can specify the preferred order by referencing another file that in turn contains a specification of resources in their preferred order.

Each resource specifier designates a network resource and a protocol specifier. A plurality of different resource specifiers can be placed on different
 15 lines of a reference file. When player 124 opens and reads a reference file, it responds by repeatedly attempting to establish a streaming data connection using the different resource specifiers in the preferred order specified by the reference file until a streaming data connection is successfully established.

A basic reference (ASX) file contains at least the URL of some
 20 multimedia content on a server. A complex link file may contain multiple files or streams arranged in a playlist, instructions on how to play the files or streams, text and graphic elements, and hyperlinks associated with elements on the media player interface.

An example of an ASX file is shown in TABLE I below.

	TABLE I -- Sample ASX File
1	<ASX version ="3.0">

	TABLE I -- Sample ASX File
2	<Title>Titan AE</Title>
3	<Abstract>Example call for Broadstream customers</Abstract>
4	<MoreInfo href = "http://www.microsoft.com"/>
5	<Entry>
6	<Title>Titan AE</Title>
7	<Author>www.reelstream.com</Author>
8	<Copyright>1999, Paramount Pictures, Inc.</Copyright>
9	<Abstract>preview from Titan AE</Abstract>
10	<MoreInfo href = "http://www.reelpreview.com"/>
11	<Ref href="http://fusion.broadstream.com/reelpreview/Calling.asp?filename=Titan_AE.asf"/>
12	</Entry>
13	<Entry>
14	<Title>Titan AE</Title>
15	<Author>www.reelstream.com</Author>
16	<Copyright>2000, Paramount Pictures, Inc.</Copyright>
17	<Abstract>ending credit for Titan AE</Abstract>
18	<MoreInfo href = "http://www.reelpreview.com"/>
19	<Ref href="http://fusion.broadstream.com/reelpreview/Calling.asp?filename=Titan_AE_Credit.a sf"/>
20	</Entry>
21	</ASX>

The various elements of ASX files are known in the art and can be found, e.g., in

http://msdn.microsoft.com/library/psdk/wm_media/wmpsdk/mmp_sdk/overview/

5 ASX_intro.htm, which is incorporated herein by reference. However, some elements are explained here for convenience.

The *Entry* (“<ENTRY>”) element of the file (e.g., lines 5 and 13 in **TABLE I**), with its associated attributes, defines to the player mechanism 124 meta-information for a single, logical piece of content (called a clip). Elements that are defined within an *Entry* element are displayed by the player mechanism 124 in a particular information area of the display panel called the Clip information area.

10 A playlist is created by stacking multiple entries. Each *Entry* element is played by the player mechanism 124 in the order they appear in the file as though the user had manually opened each clip.

The *Ref* element (e.g., lines 11 and 19 in TABLE I) specifies a URL for a content stream. The URL can point to any supported media type, using any protocol supported by the player mechanism 124. The *Ref* element is commonly used for server or protocol rollover, where, if the player mechanism 124 cannot
5 access media defined in a *Ref* element, it tries to access the URL in the next *Ref* element.

Preferably the ASX file produced by the redirection manager 143 is customized for the particular requesting client 102 based on client information which the redirection manager 143 either has or can obtain from the client 102.

10 Amongst other things, the returned ASX file points the client's browser 103 to the requested streaming media content on one of the CDNs 146. As noted above, some or all of the CDNs 146 hold the same streaming media content to provide a degree of fault tolerance and data redundancy to the system. In some embodiments, only some of the CDNs 146 will hold the same content. The ASX
15 file may contain pointers, e.g., in the form of URL's, to more than one content.

The process of a client 102 obtaining a reference file is summarized with reference to FIGURE 3. First, the Client 102 (*ClientId*) selects streaming content (*ProgramId*) from Web Site 110 (at 300) by selecting a hyperlink on the web site. The Client 102 is redirected to the Redirection Manager 143 at the Redirection
20 Server 142 (at 302). If needed (e.g., this is a first-time client), the client object wrapper (monitor) module 162 downloaded automatically to the client 102 (at 304). The redirection manager 143 then passes the request for the event to a reference file generator (at 306) which generates a reference file based on certain information including the *ClientId* (to obtain information from the Client
25 Database 152 directly or via the personalization server 150), the *ProgramId* and

available program events (from Program Event database 164), advertising (from Advertising Database 158) and available, preferably best available, CDNs 146 (provided by the bandwidth manager 166) (at 308). Then the generated reference (e.g., ASX) file is returned to the client 102 (at 310).

5 In preferred embodiments of the present invention, the reference file includes a list of CDNs from which the content is available.

Once the client 102 has received the reference file from the redirection manager 143, the client's player mechanism 124 begins playing the content referred to by the reference file, preferably under the control of the wrapper 162.

10 When the client system 102 accesses the streaming media content on one of the CDNs 146, the appropriate CDN 102 begins delivering content to the client 102 as is known in the art and the client renders (e.g., plays and/or displays) the content using the appropriate (built-in or plugin) player mechanism 124.

15 While the client 102 is receiving streaming media content from the CDN 146, the client's object wrapper 162 continuously monitors the received network traffic to evaluate quality of service (QOS) at the client 102 by measuring, e.g., network reliability, network congestion and network quality. In some preferred embodiments, the object wrapper 162 monitors the received network traffic for such measures as total bytes received, number of lost packets, number of
20 recovered packets and the like. The client object wrapper 162 sends network status information to the monitoring manager 145 at regular intervals.

The monitoring manager 145 processes all information received from clients, as described below and with reference to FIGURE 6. Some or all of the information is stored in various databases, including in the real-time logging

database 160, the client database 152, the CDN database 156, and the real-time bandwidth distribution database 154.

In a presently preferred embodiments of this invention, three types of information are sent from the client 102 to the monitoring manager 145, namely network status information, user action information and so-called ping information. Along with whatever information is sent, the client object wrapper 162 also transmits an indication (e.g., a tag) indicating to the monitoring manager 145 what kind of data/request is being sent.

The network status information is sent from the client to the monitoring manager 145 regularly, preferably at fixed intervals. In one preferred embodiment network status information is sent every two (2) minutes, and includes such information as the number of bytes transferred, the number of packets sent and/or received and/or recovered, measures of transmission quality and other types of network information. In some embodiments, the network status information includes the following items:

Network Status Information	Meaning
CmdCode = 0x0001	Code indicating type of information for use by monitoring manager
ClientIdCode	The unique identity of the client
ProgramId	The unique identity of the program being view by the client
StartTime Tick	
<i>baseURL</i>	BandWidth
<i>bufferingCount</i>	Rate
<i>bufferingClient</i>	Current Position
ClientId	
ConnectionSpeed	
Duration	
FileName	
IsBroadcast	
LostPackets	
ReceivedPackets	
ReceptionQuality	
RecoveredPackets	

When the monitoring manager **145** receives a message/request from a client **120** (from the client's wrapper **162**) (at **700** in **FIGURE 7**), the monitoring manager determines what type of message it has received (at **702**) using the tag or code *CmdCode* sent with the message. If the message tag indicates that the message is network status information (*CmdCode* = *0x0001*), the monitoring manager **145** determines that it has received real-time logging information which it stores in the real-time logging database **130** (at **704**).

The user action/event information sent to the monitoring manager **145** includes data on acts performed by a user at the client **102**. Whenever a user at the client **102** performs an event such as plays video, stop, rewind, fast forward and so forth, this user event is transmitted to the monitoring manager **145**. In some preferred embodiments, the user action/event information includes:

User Action/Event Information	Meaning
CmdCode = 0x0008	Code indicating type of information for use by monitoring manager
ClientidCode	
Programid	
StartTime	
EventCode	Play (0x0001) Stop (0x0002) FastForward (0x0004) FastReverse (0x0008) Next (0x0010) Previous (0x0020) Pause (0x0040) Cancel (0x0080)

When the monitoring manager **145** receives user action / event information (*CmdCode* = *0x0008*) from a client (at **700**), it ... (at **706**).

The monitoring manager **145** saves all information into the database. The saved information may be data mined at later time to generate various reports, e.g., for customers.

The ping information is sent to the monitoring manager 145 regularly, in some embodiments, preferably every thirty (30) seconds. The ping information is used to allow the monitoring manager 145 to know that the client object is still connected and alive.

Ping Information	Meaning
CmdCode = 0x0002	Code indicating type of information for use by monitoring manager
ClientidCode	
<i>GetCurrentEntry()</i>	
<i>CurrentPosition</i>	

5

When the monitoring manager 145 receives a *Ping* message (*CmdCode* = 0x0002) from a client 120 (from the client's wrapper 162) (at 700), the monitoring manager uses the ping information to map bandwidth distribution (at 708) and stores the associated information in the real-time bandwidth distribution database 124 (at 710).

10

If the client object monitor (object wrapper 162) detects a problem with network quality, it initiates a CDN switch-over with the monitoring manager 114 in order to link to another CDN 146 for the content.

In addition, the client object monitor 162 automatically restores services when streaming or downloading becomes interrupted because of a network problem. The client object monitor 162 sends a switch-over request message to the monitoring manager 114 requesting permission to switch to another CDN for uninterrupted download. A switch-over request message from the client object monitor 162 preferably includes the following:

15

Switch-over Request	Meaning
CmdCode = 0x0004	Code indicating type of information for use by monitoring manager
ClientIdCode	
ProgramId	
StartTime	

Switch-over Request	Meaning
<i>BaseURL</i> CDN_ID	<i>the URL of the current CDN</i> CDN ID is a number that is used internally to identify a CDN.

In response to a switch-over request from a client 102 (*CmdCode* = *0x0004*), the monitoring manager 114 evaluates the real time network status and selects a new CDN based on such factors as availability, reliability and cost-effectiveness. More specifically, the monitoring manager 145 first checks the current load balancing and distribution (at 712), and then generates a CDN switchover command/message (at 714). The identity of the new CDN is provided to the client 102 from the monitoring manager 114 in the form of a CDN switch-over reply message (at 716).

A CDN switch-over reply message preferably includes the following:

Switch-over Reply	Meaning
<i>CmdCode</i> = 0x8001	Code indicating type of information for use by client wrapper object
<i>ClientIdCode</i>	
<i>fSwitchOver</i> (1/0)	
<i>BaseURL</i> CDN_ID	<i>the URL of the new CDN to be switched to</i> CDN ID is a number used internally to identify a CDN.

Upon receipt of a CDN switch-over reply message from the monitoring manager 114, the client 102 attempts to switch to the new CDN (represented by the *baseURL* field of the switch-over reply message).

After the client 102 switches over to the new CDN, the client wrapper object 162 sends a report to the monitoring manager 114 to that effect.

If the client wrapper object 162 is unable to connect to the monitoring manager 114 in order to initiate a switch over, the client may initiate a self-controlled or self-directed CDN switchover. The reference file will preferably include a list of CDNs from which the content is available. If a client needs to

make a self-directed CDN switchover, the client needs only to select one of the CDNs from the list in the reference file.

Preferably a CDN switchover, whether client or manager initiated, causes little or no disruption to the content being displayed. However, since the content
5 is being streamed to the client from the original CDN, the new CDN needs to begin its content streaming at or substantially close to the current position of the current streamed content.

When the client wrapper object **162** determines that the quality of the
10 stream from the CDN **146** has degraded to an unacceptable level, e.g., there are too many lost packets or the like, the client wrapper object **162** preferably sends a CDN change request to the monitoring manager **114**. Upon receiving such a request from a client system, the monitoring manager **114** chooses a new CDN
146 to supply the streaming media content to the client as described above. The
15 monitoring manager **114** then sends a message to the viewer system (the client **102**) identifying the newly-chosen CDN. The client **102** polls its media player to identify the streaming media which is being played, as well as a time index thereof, e.g., how far the playback is into the stream. The client **102** then advances this time index a predetermined amount of time and sends a request to
20 the new CDN server asking it to begin sending a stream from the identified content to the client at that time. Also, at that advanced time, the client **102** directs its streaming media player to terminate its stream with the old CDN. In this way, reception and playback of the streaming media continues virtually interrupted by the changeover to the new CDN. Currently, this predetermined
25 time is around 7-8 seconds. That includes about two to three (2-3) seconds of

connection time and about five (5) seconds of buffering. These timing numbers assume good network conditions.

A preferred way to do a seamless switch over (i.e., perceptibly instantaneous switchover) is as follows. A first Windows Media Object plays on the screen, and a second instance of the Windows media player object is instantiated behind the first one. Both Windows media player objects are synchronized by starting the second Windows Media Object and setting its time mark (current entry and current position variable) equal to that of the first Windows Media Object. After the second Windows Media Player finishes buffering and starts to stream, the first Windows Media Player is killed. Thus creating an effect of instantaneous CDN switch over.

The monitoring manager 114 can also initiate a change in the CDN 146 associated with a particular client 102. It may do so, for example, when scheduled maintenance is to be performed on one of the CDNs 146. At a predetermined time in advance of the beginning of the maintenance period, the monitoring manager 114 sends commands to all clients 102 receiving streams from that CDN directing them to begin receiving the streaming content from selected other ones of the CDNs 146. To do so, the monitoring manager 114 first selects new CDNs for each of the affected clients, sends CDN switchover commands to each such client advising it of its new CDN. The affected clients affect CDN changeovers as described above. Presently preferred embodiments use HTTP instead of TCP/IP to communicate with the monitor. This allows clients to communicate with the monitoring software through a standard HTTP port. This also means that the CDN switch over command can be issued during the PING cycle.

In addition to being used for changeover purposes, logging information such as the above is sent to the monitoring manager 114 via the Internet 104. The monitoring manager 114 stores this logging information in a client information database 120. The monitoring manager 114 puts all real time network data into the database. It also saves history of network information in the database. The monitoring server can at anytime look at the status of the multiple CDN network and measure network quality and network traffic. The network will adjust itself to balance itself as problems arises. Additionally, information stored in this database for a particular content provider is available for viewing in real time by that content provider via a web page on the web server. Preferably, system administrators can view all logging information as well as statistics derived therefrom.

In summary, with reference to the state diagram in FIGURE 4, a particular client 102 is initially in a start state (S400). As described above, the client 102 requests a reference file from the redirection manager 143. When the ASX file is returned to the client 102, the client connects to the appropriate CDN 146 and the client's player mechanism 105 begins to play the streamed content (at state S402). While the streamed content is being played, the wrapper 162 monitors the system. If the wrapper determines that the network has failed or otherwise determines that the network performance is unacceptable (e.g., there is network congestion), the wrapper 162 moves to a "*network bottleneck or CDN switchover*" state (S404). The wrapper 162 notifies the monitoring manager 145 about the problems and waits for instructions. While waiting for instructions from the monitoring manager 145, the client's player mechanism 105 continues to play the content, if possible (i.e., if there has not been complete network failure).

If the wrapper **162** does not receive instructions from the monitoring manager **145** within a predetermined amount of time (preferably less than X 5- 10 seconds), or if the client's wrapper **162** cannot establish a link to the monitoring manager, the client initiates a self-controlled CDN switchover (in state **S406**). On the other hand, if the wrapper **162** receives timely instructions from the monitoring manager **145**, the client initiates a monitor-controlled CDN switchover (in state **S408**). In either case, when the switchover has taken place, the client's system returns to the play state (**S402**). When the stream ends, the system enters the "*Done*" state.

The Redirection & Bandwidth Managers

Upon a client request for streaming content (at **300** in **FIGURE 3**), the redirection manager **143** must determine which CDN **146** the requesting client should use. The decision by the redirection manager **143** as to which CDN to use is made preferably according to predetermined rules, e.g., weighing factors such as network load, reliability and cost effectiveness.

The redirection server **142** includes a bandwidth manager mechanism **128** which provides the redirection manager **143** with information needed to make its decisions. As shown in **FIGURE 5**, the redirection manager **143** periodically asks (at **500**) the bandwidth manager mechanism **128** how to weight each CDN **146** per a particular content distributor. In some preferred embodiments, the redirection manager **143** asks the bandwidth manager mechanism **128** for information at least every ten (10) minutes. In some preferred embodiments, the redirection manager **143** asks the bandwidth manager mechanism **128** for information when needed in addition to periodically.

In response to the request from the redirection manager 143 (or on an ongoing basis) the bandwidth manager mechanism 128 determines (at 502) how to balance the load, minimize the cost and maximize the performance of the system 100. Accordingly, the bandwidth manager mechanism 128 queries the client database 122 (at 504) to determine which CDNs 146 are working with this distributor (customer). In addition, the bandwidth manager mechanism 128 queries the CDN database 126 (at 506) to determine the cost and pay structure per CDN 146. The bandwidth manager mechanism 128 also queries logging information (at 508) in the real-time logging database 130 to evaluate recent network reliability. Having obtained the available information, the bandwidth manager mechanism 128 calculates (at 510) how to balance the load, minimize the cost and maximize the performance of the system 100 and provides the redirection manager (at 512) with information needed to select an appropriate CDN 146 for a particular client and content selection.

Once the redirection manager 143 has determined (at 514) which particular CDN 146 the client 102 is to use, the redirection manager 143 then returns information back to the client 102 so that the client can immediately view the requested content.

Preferably the redirection manager 143 selects the lowest cost CDN 146 for any particular client and content.

Advertising

In some embodiments of the present invention, when the redirection manager 143 creates the ASX file for a particular client, advertising information

(or other content) is added to the ASX file. The advertising may be directed to the client based on the client's identity (*ClientId*) and other information.

As ASX file or the streamed content can include instructions to the player mechanism 105 to cut away from a stream and to play other streams or files according to scripting in the file. For example, during a live Internet broadcast of an event, a script command can be sent at the beginning of every commercial break that instructs each client to play commercials listed in their respective ASX files. When clients finish playing the commercials, scripting in the metafile instructs each client to cut back to the live broadcast. Advertising insertion is preferably implemented using the *Event* element.

FIGURES 8(a) and 8(b) show the inclusion of advertisement in different types of streaming content delivery according to embodiments of the present invention. Specifically, FIGURE 8(a) shows the processing real-time ad insertion in regular (scheduled) programs, whereas FIGURE 8(b) shows that processing for real-time ad insertion for live events.

With reference to FIGURE 8(a), when a client 102 requests streaming content (i.e., requests a reference file) that plays a scheduled event (at 800), the redirection manager returns to the client (at 802) a reference file that, when played by the client's player 124, will play that event. The client's web browser asks the ad manager 174 what stream is to be played with this event (at 804). In response to this request, the ad manager 174 queries and gets information from the advertising database 158 and from the personalization server 150 (at 806). Using this information, the ad manager 174 determines which ad is to be played by the client and downloads the ad stream information to the client's web browser (at 808). The client's player 124 starts to queue the ad for play (at 810). When the

client's player reaches the next item on its play list (the ad), it plays the queued ad (at 812). Then, when the ad is done, the player 124 plays the next content item on its playlist (at 814).

With reference to FIGURE 8(b), when a client 102 requests live streaming
5 content (i.e., requests a reference file) that plays a live event (at 816), the redirection manager returns to the client (at 818) a reference file that, when played by the client's player 124, will play the requested event. Unlike the case of scheduled events, where the playlist has markers for the insertion of ads, in the case of a live event, the ad manager issues an event call to the client when an ad is
10 to be played (at 820). In response to such a call, the client asks the ad manager 174 what stream is to be played with this event (at 822). In response to this request, the ad manager 174 queries and gets information from the advertising database 158 and from the personalization server 150 (at 824). Using this information, the ad manager 174 determines which ad is to be played by the client
15 and downloads the ad stream information to the client's web browser (at 826). The client's player 124 queues the ad for play (at 828), stops playing the live event and then plays the queued ad (at 830). Then, when the ad is done, the player 124 continues with the live event (at 832).

In both of the cases described above, the ad is preferably selected based on
20 client (personal) information, the ad spot and the content being viewed. Preferably an ad is not selected until the spot has been sold and the ad is then served in real-time with an Event call to the client. Normal programming is resumed after the advertisement call.

Example of Operation

An example of typical operation of embodiments of this invention is presented here, by way only of example.

Real Time Load Balancing With Real Time Feedback

5 **FIGURE 9(a)** generally shows real time load balancing with real time feedback according to the preferred embodiment of the present invention. This example assumes that the client has used the system before and so does not need an initialization process. That is, in this example, it is assumed that the client wrapper/monitor mechanism **162** is present on the client's system. As shown in

10 **FIGURE 9(a)**, first a user/client requests a particular video (at **900**). This is done, e.g., by the client selecting a hyperlink for that video on a particular web site. The hyperlink directs the client to the redirection manager **143** and the client provides the redirection manager **143** with the client's identity and the program identity for the desired video. The redirection manager **143** creates a reference (e.g. ASX) file

15 for the requested video and returns the reference file to the client (at **902**). The client's player **105** then begins to play the video (at **904**) while, at the same time, the client wrapper **162** starts to monitor the system (at **906**). If the wrapper **162** determines that there is network congestion (or failure) (at **908**), then the client contacts the monitoring manager **145** (at **910**), reports the problem and asks for a

20 monitor-controlled CDN switchover. The client then waits for a response from the monitoring manager **145**. While waiting for a response from the monitoring manager **145**, the player **124** continues playing, if possible (i.e., if the connection to the CDN has not been completely lost and/or the player **124** has some content buffered).

If no network congestion was detected (at 908) by the client wrapper 162, then, if scheduled or necessary, the wrapper 162 reports any information (e.g., ping, status, etc.) and/or user events to the monitoring manager 145 (at 912) and processing continues. If all of the requested content referred to in the reference
5 (ASX) file has been completed by the player 124 (at 914) then processing is done (at 916), otherwise playing and monitoring continue (at 904 and 906).

If the client detects network congestion (at 908) but is unable to contact the monitoring manager 145 or the wait for a request to switchover to another CDN times out (at 918), then the wrapper/monitor 162 initiates a self-controlled
10 CDN switchover (at 920) and then processing continues (at 914), if there is any content remaining to be played. In the case of a self-controlled CDN switchover, the player 124 switches over to another CDN specified in the reference file.

On the other hand, if the client determines that there is congestion (at 908) and is successful in contacting the monitoring manager 145 (at 910) and does not
15 time out, the client obtains new CDN information from the monitoring manager 145 (at 922) and initiates a CDN switchover (at 924) to the new CDN. If the program has been completed (at 914) then processing is done (at 916), otherwise playing and monitoring continue (at 904 and 906).

20 The various mechanisms described herein, including, without limitation, the monitoring manager, the redirection manager, the bandwidth manager and the wrapper / monitor mechanism may be implemented in hardware, software or a combination thereof. When implemented in software, they may be implemented in any type of appropriate interpreted or compiled programming language. In
25 preferred embodiments of this invention, the wrapper mechanism is implemented

in the machine-independent Java™ programming language as small, powerful and fast C++ ATL COM objects. When implemented fully or partially in software, aspects of the invention can reside on any memory or storage medium, including but not limited to a ROM, a disk, an ASIC, a PROM and the like.

5 When the various mechanisms of the present invention are running on a particular machine (e.g., the at the client or on a server), they may reside in the memory of the machine or on a storage device or in a combination.

 While the invention has been described with reference to particular mechanisms (algorithms, processes and functions) and architectures, one skilled
10 in the art would realize that other mechanisms and/or architectures could be used while still achieving the invention.

 While embodiments of the present invention have been described with particular setup and initialization procedures, other setup and/or initialization procedures can be used.

15 Further, while many of the operations have been shown as being performed in a particular order, one skilled in the art would realize that other orders, including some parallelization of operations, are possible and are considered to be within the scope of the invention.

 The present invention has been described above in connection with a
20 preferred embodiment thereof; however, this has been done for purposes of illustration only, and the invention is not so limited. Indeed, variations of the invention will be readily apparent to those skilled in the art. For example, the redirection and monitoring servers may be combined into a single server, or the system may additionally be used to store content developer's streaming media
25 content. Such variations also fall within the scope of the invention.

What Is Claimed Is:

1. A system for delivering streaming media over a distributed communication network comprising:
 - a plurality of content distribution networks (CDNs) for delivering the streaming media to a plurality of clients;
 - load balancing means for evenly distributing a delivery load imposed by the plurality of clients over the plurality of delivery means; and
 - client wrapper means, in each client, for providing information to the load balancing means for use in distributing the delivery load.
2. A system as in claim 1 wherein each client wrapper means comprises:
 - means for monitoring network traffic received at the client to evaluate quality of service at the client.
3. A system as in claim 2 wherein the means for monitoring comprises:
 - means for measuring at least one of network reliability, network congestion and network quality.
4. A system as in claim 3 wherein the means for monitoring monitors the received network traffic for at least one of: (a) total bytes received, (b) number of lost packets, and (c) number of recovered packets.

5. A system as in claim 2 wherein the means for monitoring sends to the load balancing means at least one of (a) network status information; (b) user action information; and (c) client status information.

6. A system as in claim 5 wherein the network status information is sent to the load balancing means regularly.

7. A system as in claim 6 wherein the network status information is sent at least every minute

8. A system as in claim 6 wherein the network status information includes at least some of: (a) a number of bytes transferred to the client, (b) a number of packets sent and/or received by the client; and (c) a number of packets recovered by the client from multiple CDNs.

9. A system as in claim 1 wherein the client wrapper means comprises:

means for detecting a problem with network quality; and
switchover means for, when a problem is detected with network quality, initiating a CDN switchover.

10. A system as in claim 9 wherein the switchover means comprises:
means for requesting a CDN switchover from the load balancing means; and

means for performing a self-controlled CDN switchover in the event that a requested switchover does not take place.

11. A method of managing delivery of streaming media in a system wherein streaming content is delivered to a client from a plurality of content distribution networks (CDNs), the method comprising, at a client:

obtaining a reference file containing a reference to at least one CDN;
requesting streaming content from a CDN in the reference file; and
monitoring traffic received at the client to evaluate quality of service at the client.

12. A method as in claim 11 further comprising:
sending to a monitoring manager information regarding the monitored quality of service at the client.

13. A method as in claim 12 wherein the information sent includes at least one of (a) network status information; (b) user action information; and (c) client status information.

14. A method as in claim 11 further comprising:
initiating a CDN switchover if a problem is detected with network quality,

15. A method as in claim 14 further comprising:
performing a self-controlled CDN switchover in the event that a requested switchover does not take place.

16. A method as in claim 11 further comprising:

receiving an instruction to perform a directed CDN switchover; and

performing a directed CDN switchover.

17. A method of managing delivery of streaming media in a system

wherein streaming content is delivered to a plurality of clients from ones of a plurality of content distribution networks (CDNs), the method comprising, at a monitoring manager:

receiving status information from ones of the plurality of clients;

based on received status information, directing a particular client to switchover to a different CDN.

18. A method of managing delivery of streaming media in a system

wherein streaming content is delivered to a plurality of clients from ones of a plurality of content distribution networks (CDNs), the method comprising:

receiving a request from a particular client of the plurality of clients to obtain a particular streaming content;

selecting a particular CDN to provide the requested content to the particular client; and

providing the client with a reference to the selected CDN.

19. A method as in claim 18 wherein the selecting of the particular

CDN is based on at least some of:

(a) available CDNs;

- (b) the requested content;
- (c) the status of the network; and
- (d) pricing information.

20. A method as in claim 18 further comprising:

providing the client with a list of CDNs from which the content may be obtained.

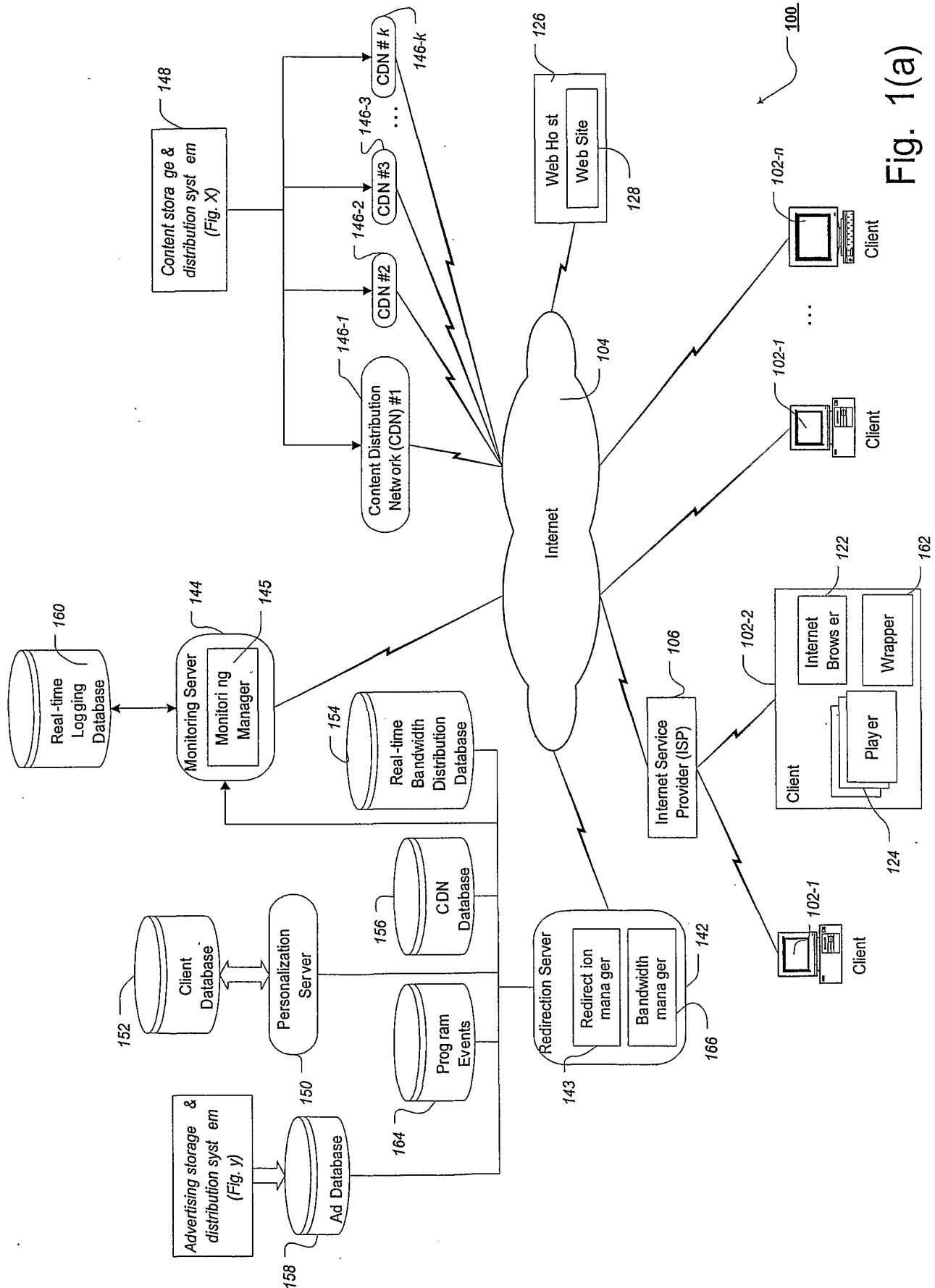


Fig. 1(a)

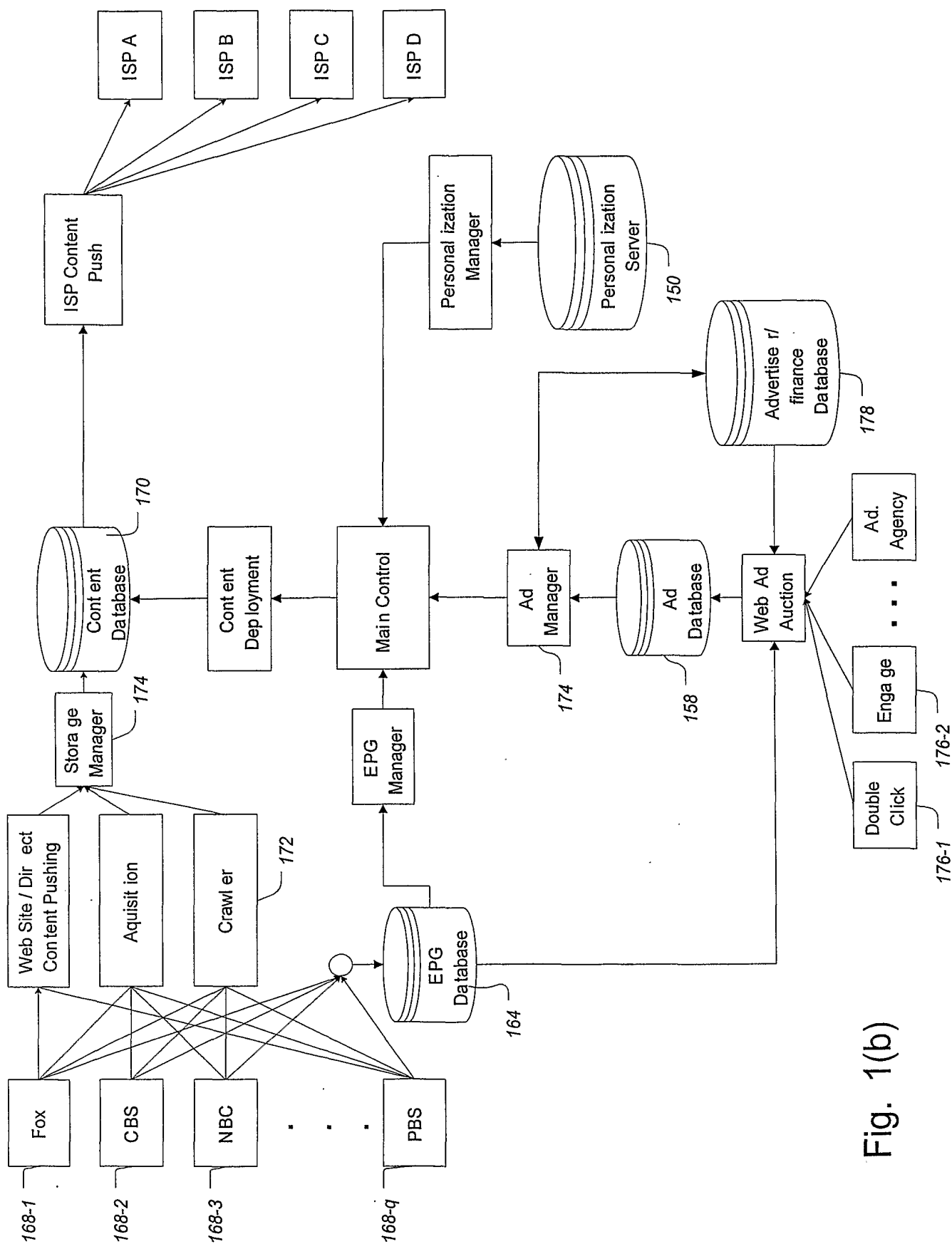


Fig. 1(b)

Fig. 2(a)

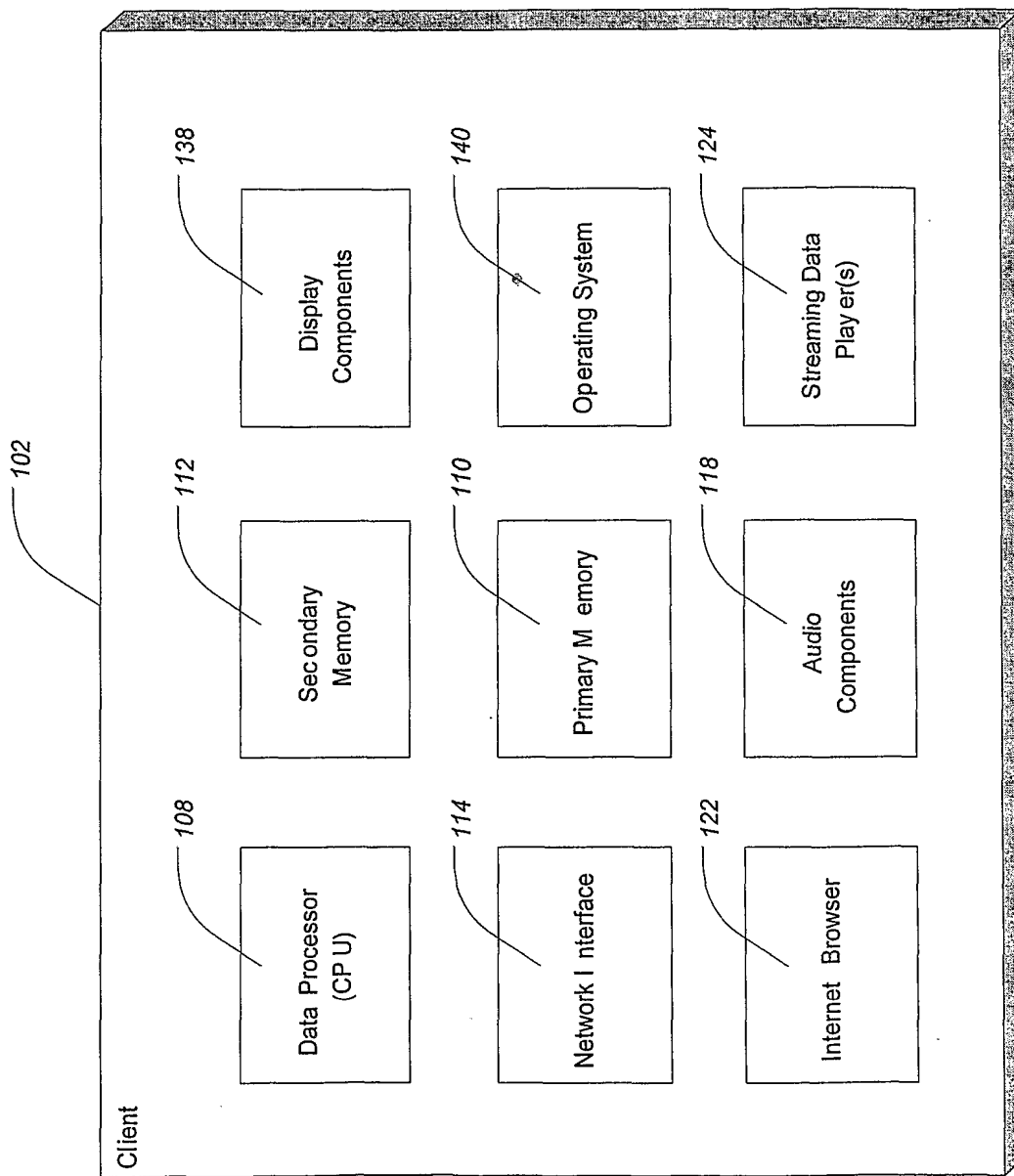
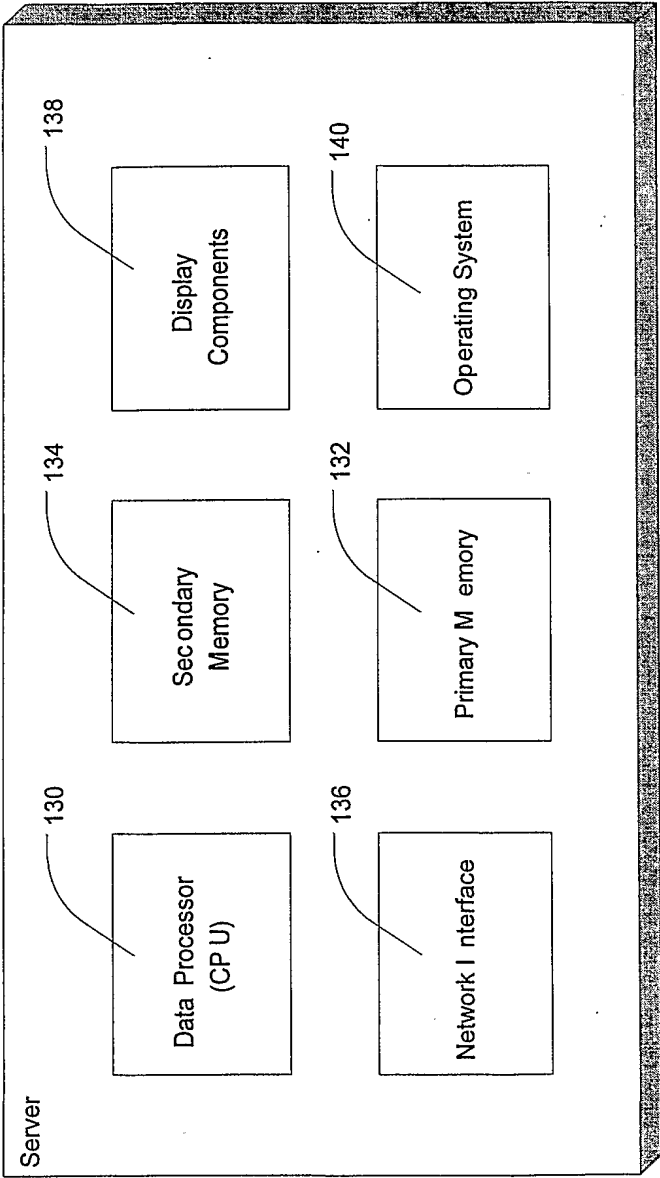


Fig. 2(b)



5/12

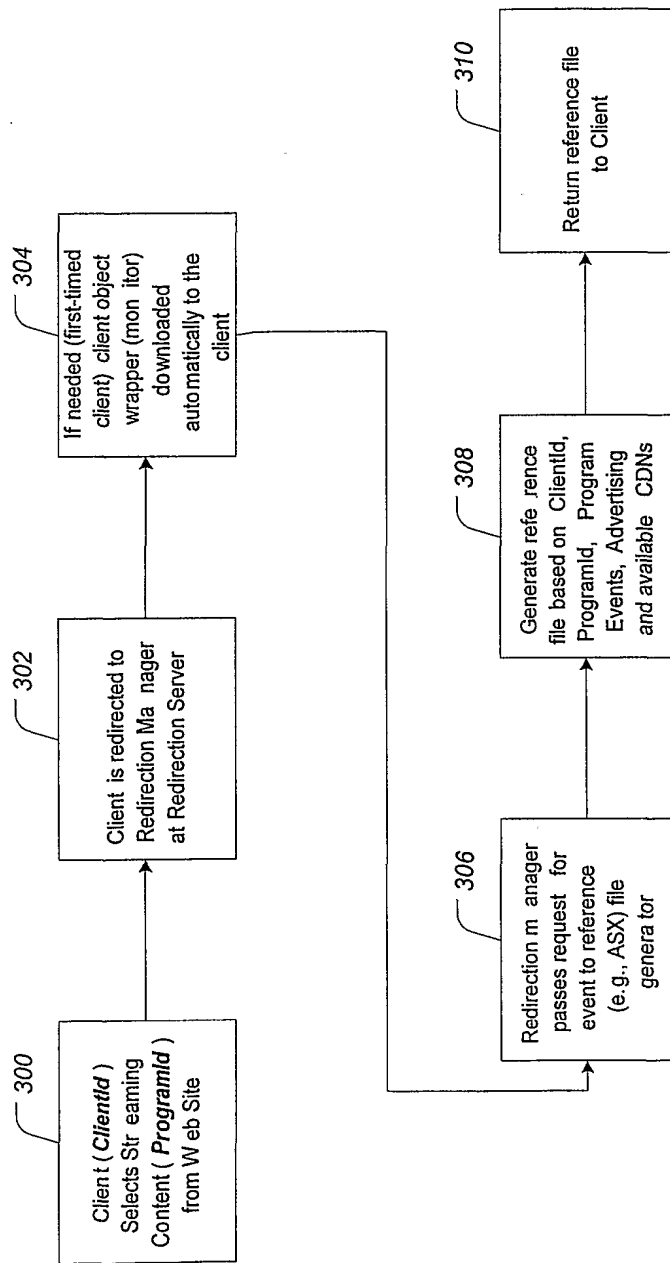
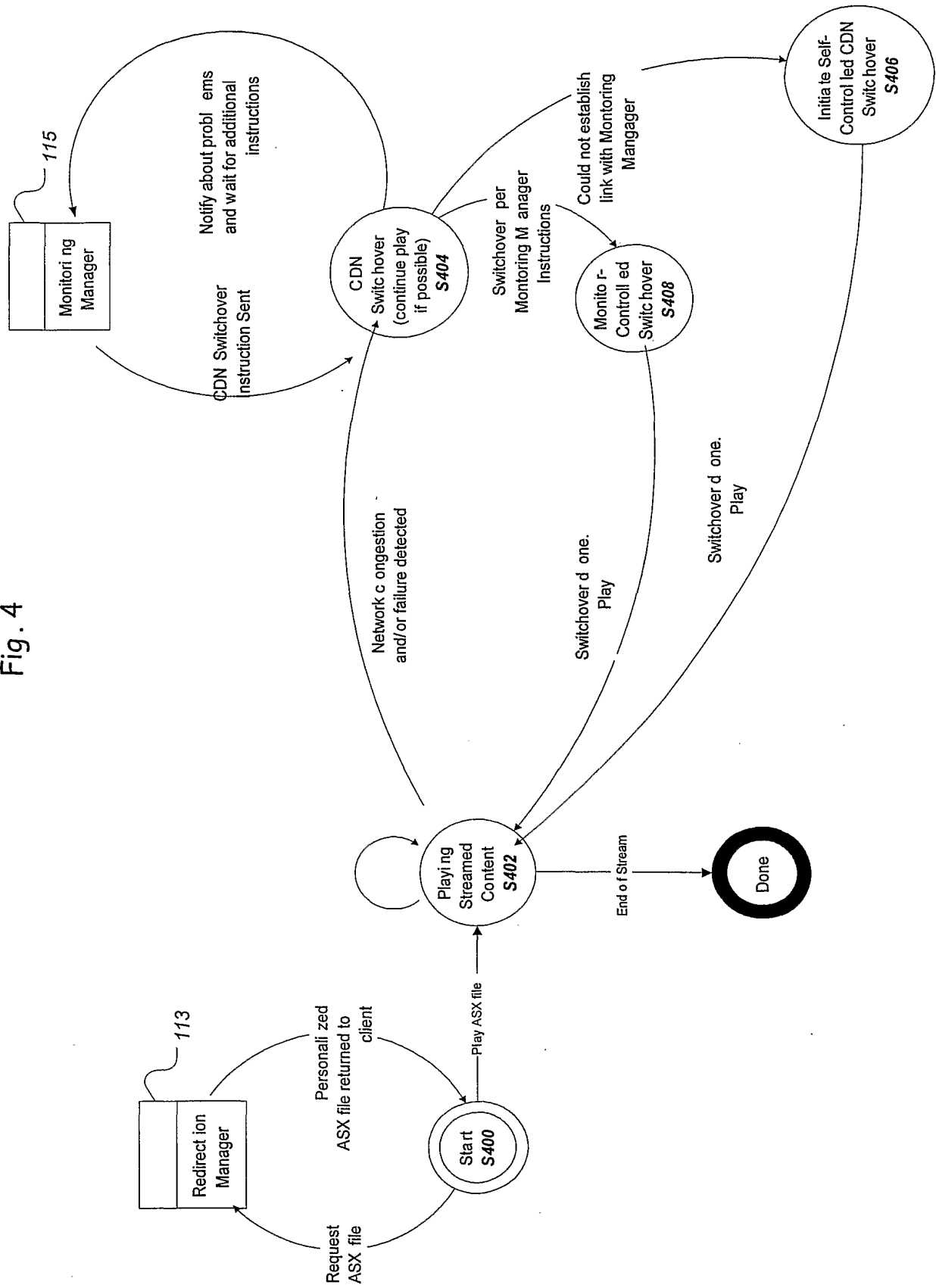


Fig. 3

6/12

Fig. 4



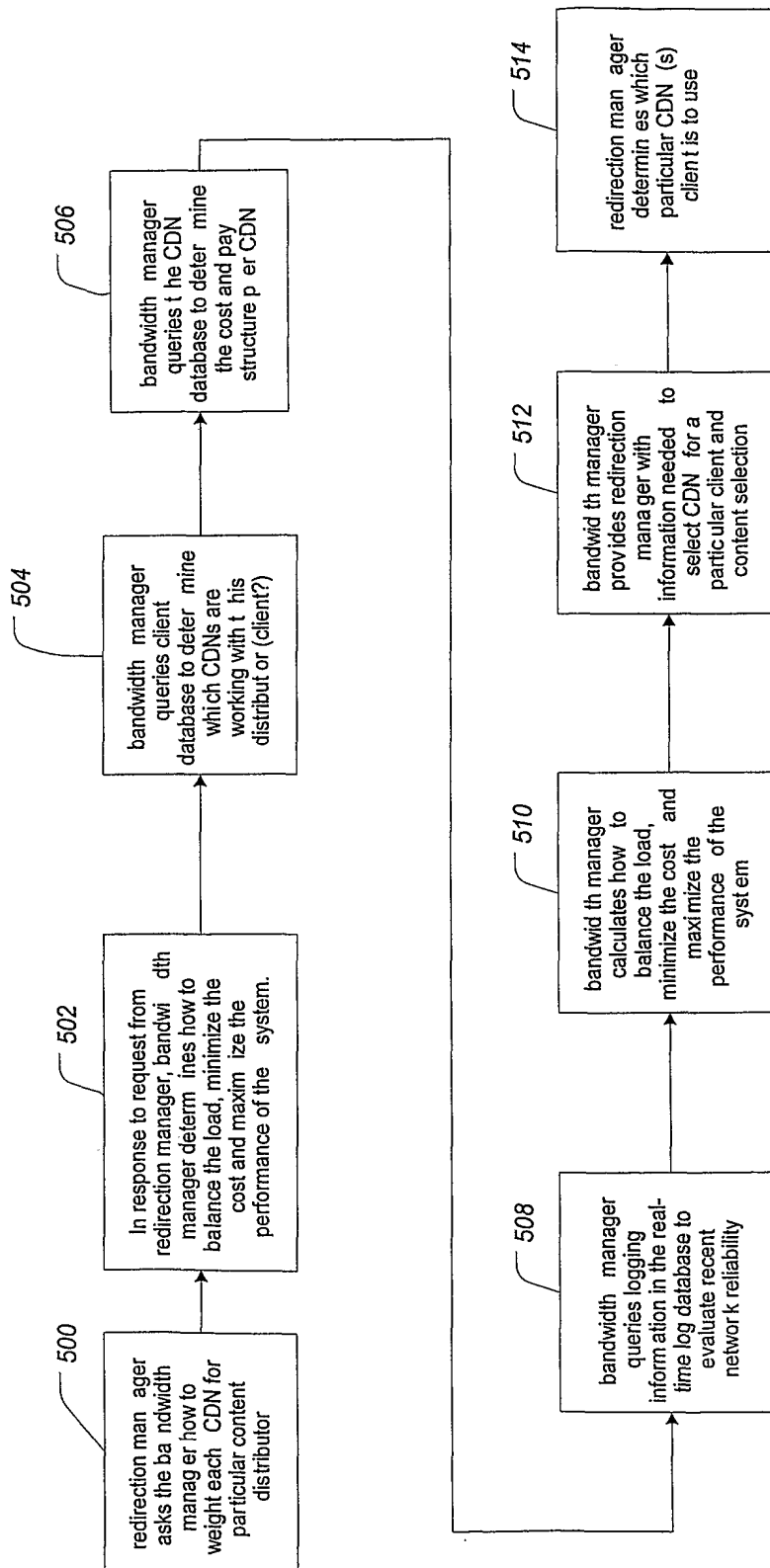


Fig. 5

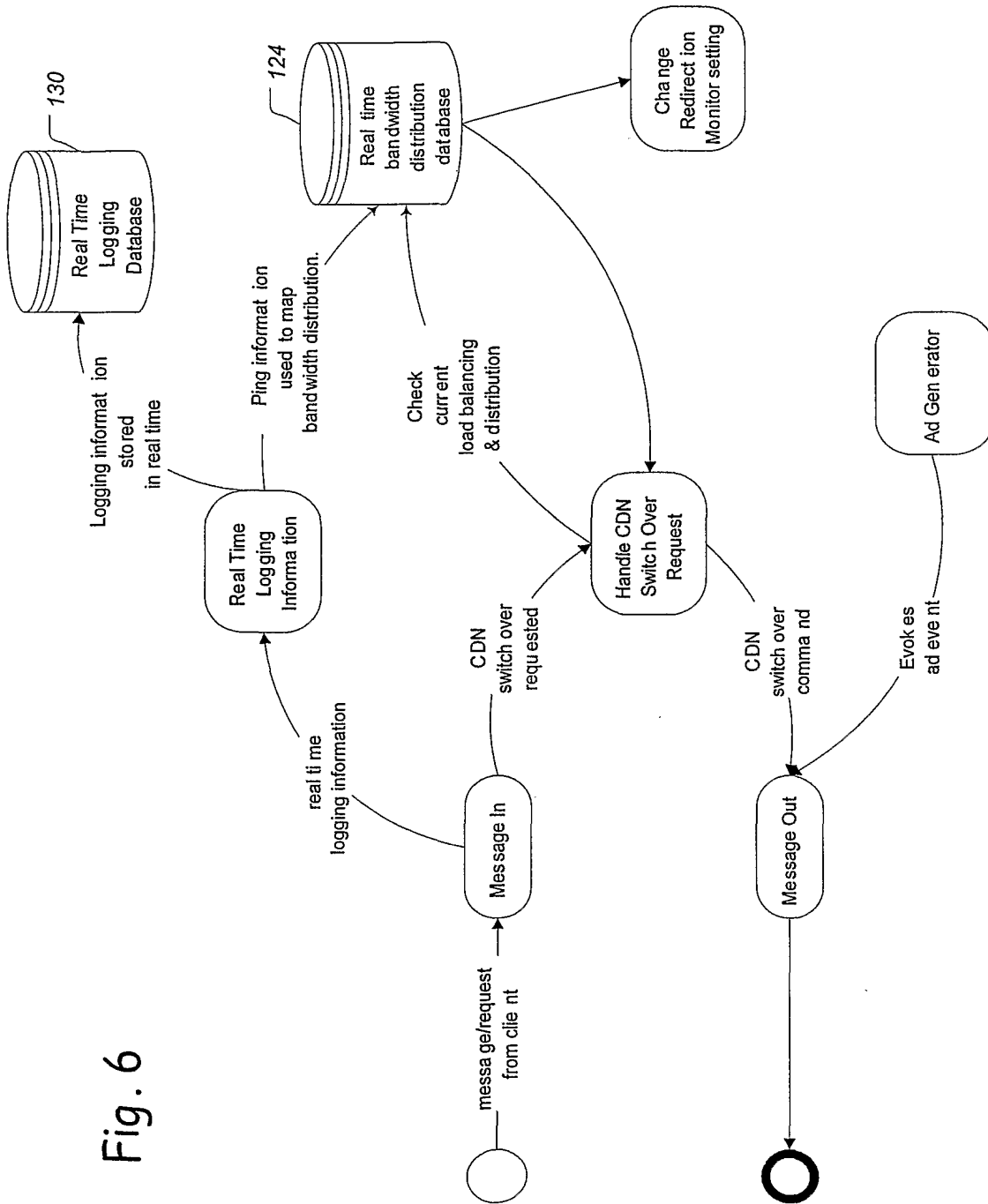
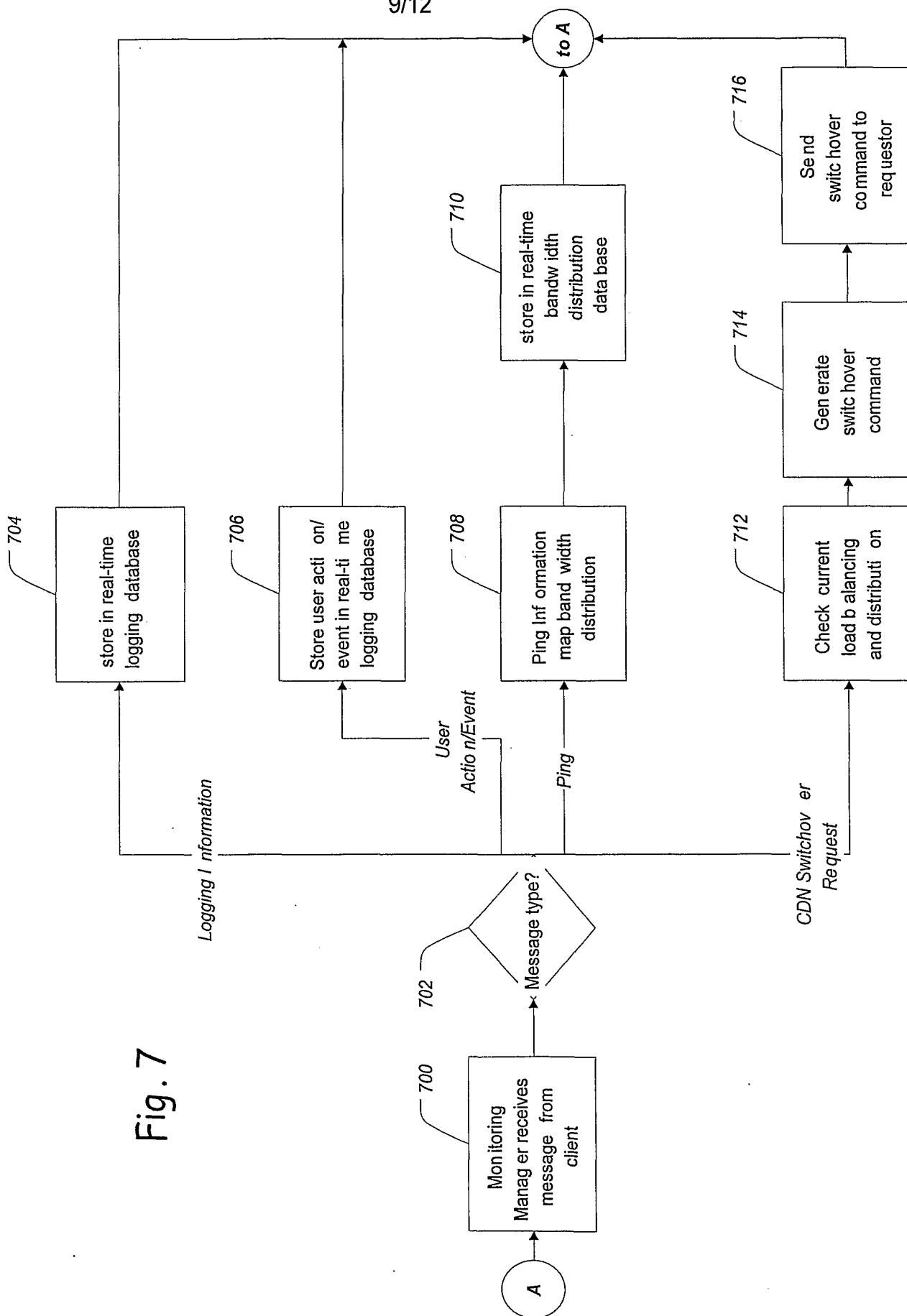


Fig. 6

9/12



10/12

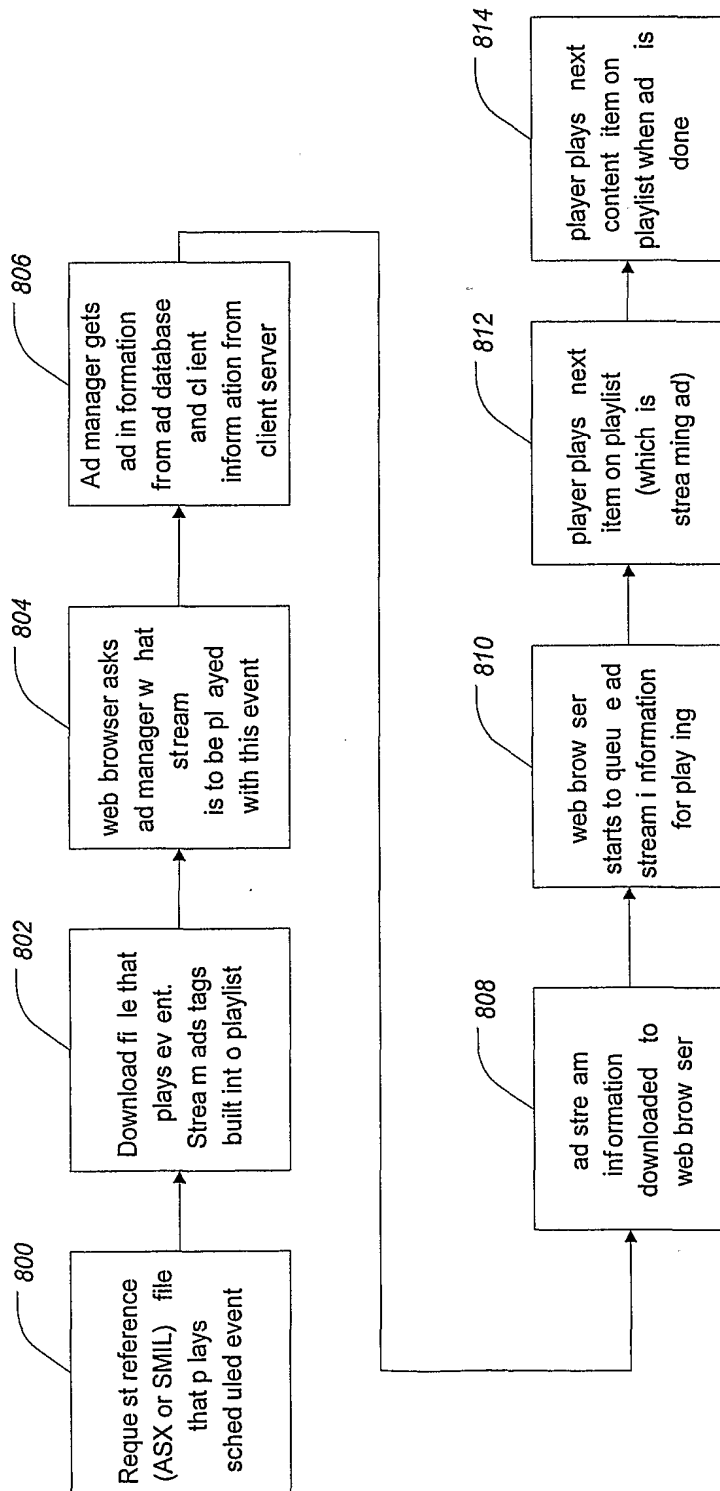
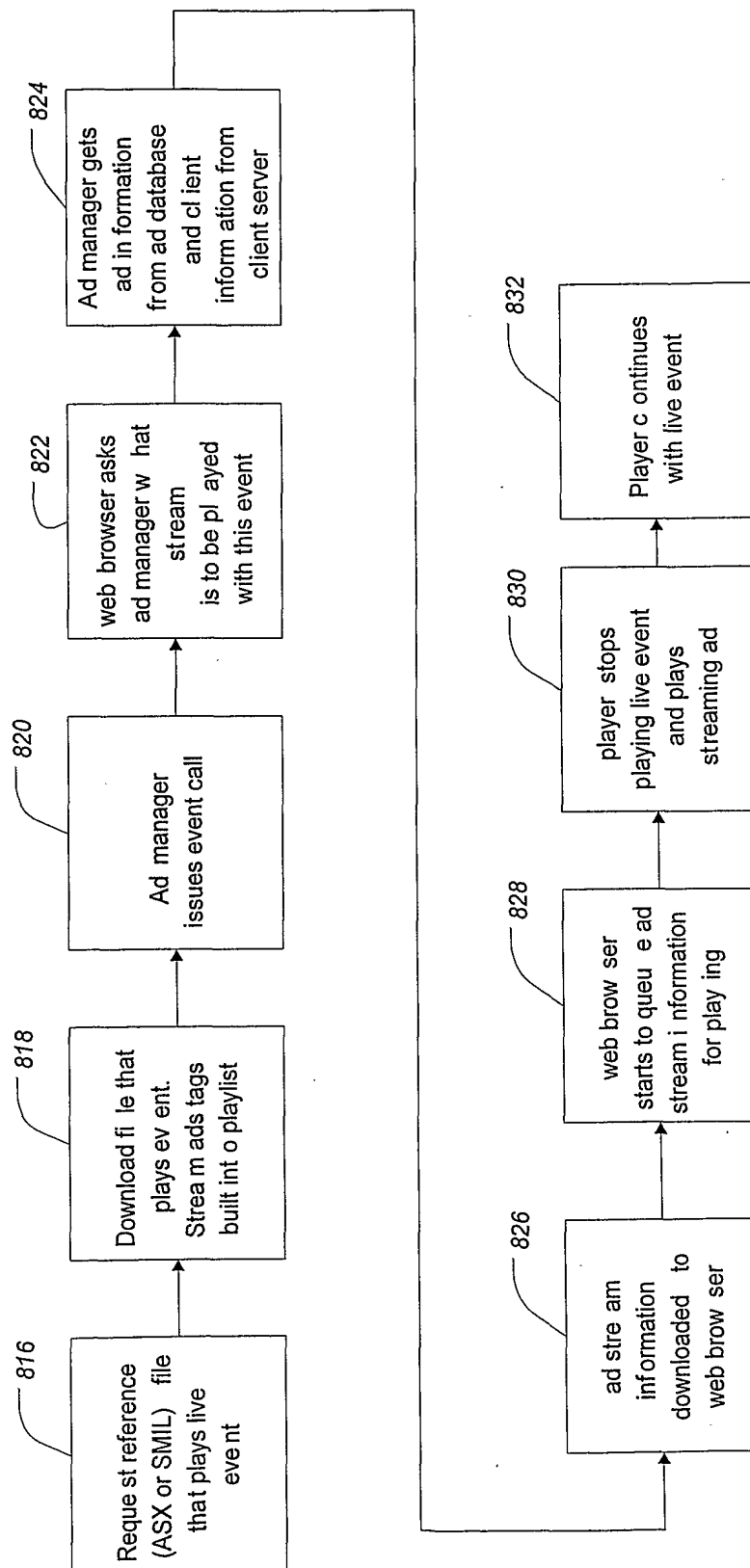


Fig. 8(a)

11/12

Fig. 8(b)



12/12

Fig. 9

